

Approximation Schemes for Capacitated Geometric Network Design*

Anna Adamaszek¹, Artur Czumaj¹, Andrzej Lingas², and Jakub Onufry Wojtaszczyk³

¹ DIMAP and Department of Computer Science, University of Warwick, UK
{A.M.Adamaszek, A.Czumaj}@warwick.ac.uk

² Department of Computer Science, Lund University, Sweden
andrzej@cs.lth.se

³ Google, Inc.
onufry@google.com

Abstract. We study a *capacitated network design problem* in *geometric* setting. We assume that the input consists of an integral link capacity k and two sets of points on a plane, sources and sinks, each source/sink having an associated integral demand (amount of flow to be shipped from/to). The *capacitated geometric network design problem* is to construct a minimum-length network N that allows to route the requested flow from sources to sinks, such that each link in N has capacity k ; the flow is splittable and parallel links are allowed in N .

The capacitated geometric network design problem generalizes, among others, the geometric Steiner tree problem, and as such it is NP-hard.

We show that if the demands are polynomially bounded and the link capacity k is not too large, the single-sink capacitated geometric network design problem admits a *polynomial-time approximation scheme*. If the capacity is arbitrarily large, then we design a *quasi-polynomial time approximation scheme* for the capacitated geometric network design problem allowing for arbitrary number of sinks. Our results rely on a derivation of an upper bound on the number of vertices different from sources and sinks (the so called Steiner vertices) in an optimal network. The bound is polynomial in the total demand of the sources.

1 Introduction

The area of *network design*, problems of designing low cost networks satisfying some predefined constraints, plays a fundamental role in operational research and graph algorithms. We consider one important class of problems in this category, the *capacitated network design problem*: for a given input network N , find a multiset of links in N of minimum cost that will allow to route a predetermined amount of flow from a set of sources to a set of sinks subject to the capacity constraints on the links.

While network design problems have been extensively studied in operational research and combinatorial optimization, traditionally the main focus has been on problems modeled by arbitrary graphs. However, as it has been observed in some more applied papers, many applications require to consider network design problems in the

* Research supported in part by the Royal Society IJP-2006/R2, the Centre for Discrete Mathematics and its Applications (DIMAP), EPSRC EP/D063191/1, and VR grant 621-2005-4085.

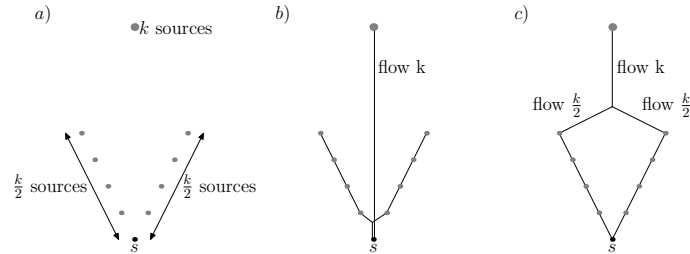


Fig. 1. A network for which the best solution which does not allow splitting the flow (b) has a significantly larger cost than the best solution which allows splitting the flow (c). All sources have unit demands and the single sink s has demand $2k$.

geometric setting. For example, when Salman *et al.* [10] initiated the graph-algorithmic study of the *buy-at-bulk network design problem* (besides the underlying graph with distinguished source and sink nodes, and their demands, there is also given a discrete set of types of links having different capacities and costs that can be used to assemble a network), they considered Euclidean graphs as a central case. Only recently the problem has been considered in the context of *geometric network design* [2], where the only input are the locations of sources and sinks, their demands, and a set of types of links.

In this paper, we study the capacitated network design problem in *geometric* setting. The input consists of an integral *link capacity* k and two sets of points on a plane, *sources* and *sinks*, each source/sink having an associated integral *demand* (amount of flow to be shipped from/to). The capacitated geometric network design problem is to construct a minimum-length network N that allows to route the requested flow from sources to sinks, such that each link in N has capacity k . We assume the flow is *splittable* and *parallel links* are allowed in N . (Observe that this problem can be considered as a special case of the geometric buy-at-bulk network design problem, in which only a single type of links having a capacity constraint is available.)

We term the studied problem as *capacitated geometric network design (CGND)* or *single-sink capacitated geometric network design (SCGND)* if there is only one sink.

It is not difficult to observe that even the single-sink variant is NP-hard, as it includes as a special case the minimum Euclidean Steiner tree problem [4]¹. Therefore, we focus on the design of fast approximation schemes.

Unlike in the Steiner tree problem, a near optimal capacitated geometric network is not necessarily a tree (see, e.g., Fig. 1). This complicates the task of deriving efficient approximation schemes for CGND and SCGND. Another major difficulty is caused by the so-called *Steiner vertices*, i.e., vertices different from the sources and sinks where the links meet and branch. Since Steiner points can be arbitrary points on the plane, one can consider all points in the vicinity of sources and sinks as potential candidates for Steiner vertices. In the very special case of minimum Euclidean Steiner tree the number of Steiner vertices in an optimal solution can be easily upper bounded by $n - 2$, where n is the number of input points. If the network edges are required to be vertical or horizontal, then the Steiner points can be constrained to the quadratic number of

¹ We can model the Steiner tree problem by the SCGND problem, by taking one of the input points as sink, all other points as sources of unit demand, and set the link capacity to $n - 1$.

intersections between vertical and horizontal straight-lines passing through the sources and sinks, the so called Hanan grid [2,11]. However, in the general Euclidean case that we consider here, the problem of bounding the number of Steiner points in terms of the total demand of input sources has been open [2].

Our contributions. We present three results for the CGND problem. First, we study structural properties of (near) optimal solutions for CGND. Our main result here is an upper bound for the number of Steiner points in an optimal solution that is polynomial in the total demand of sources; no upper bound for the number of Steiner points has been previously known. This structural result allows us to design a *quasi-polynomial time approximation scheme (QPTAS)* for CGND with polynomially bounded demands of sources and sinks. Next, we extend this result to derive our main algorithmic result, a *polynomial-time approximation scheme (PTAS)* for SCGND when the link capacity is at most $2^{O(\sqrt{\log n})}$, and the demands of sources and sinks are polynomially bounded.

The QPTAS is obtained by combining our upper bound on the number of Steiner points with Arora’s framework [1] for geometric optimization problems. The main result of the paper, the PTAS, relies on a geometric partition of the sources combined with a TSP-based heuristic and our QPTAS applied to a small number of points.

Related work on geometric network design. A related minimum Euclidean Steiner tree problem has been studied extensively in the literature (cf. [7]), with a PTAS provided independently by Arora [1] and Mitchell [8].

Salman *et al.* [10] initiated the algorithmic study of the single-sink buy-at-bulk network design problem. They argued that the problem is especially relevant in practice in the *geometric* case. They provided a polynomial-time approximation algorithm for a variant of buy-at-bulk single-sink network design on an Euclidean graph with an approximation ratio of $O(\log(D/c_1))$, where D is the total capacity of sources and c_1 is the smallest link capacity. Besides allowing the use of many link types, their model differs from ours in that they did not permit a flow to be split, and only admit a given finite set of points in the plane to be used as vertices by the network, whereas we allow splittable flow and we allow Steiner points to be arbitrary points in the plane. Czumaj *et al.* [2] considered geometric buy-at-bulk network design allowing the entire space to be used. They presented a QPTAS for the rectilinear variant of geometric network design with polynomially bounded total demand, where the links have to be horizontal or vertical. Their QPTAS relies on the observation that in this case Steiner points can be constrained to lie on the Hanan grid. Hassin *et al.* [6] and Morsy *et al.* [9] considered the special case of the network design problem raised in [10], where only one type of link occurs. Since their flow is unsplittable, the result is not comparable to our model.

2 Preliminaries

Consider the Euclidean 2-dimensional space \mathbb{E}^2 . Let $\{s_1, \dots, s_{n_s}\}$ be a given set of n_s points in \mathbb{E}^2 (*sources*), and let $\{t_1, \dots, t_{n_t}\}$ be a given set of n_t points in \mathbb{E}^2 (*sinks*). Let $n = n_s + n_t$. Each source s_i supplies some integral *demand* $\vartheta(s_i)$ to the sinks, and each sink t_j is required to receive some integral *demand* $\vartheta(t_j)$ from the sources. We assume that $\sum_i \vartheta(s_i) = \sum_j \vartheta(t_j)$ and define $D = \sum_i \vartheta(s_i)$ to be the *total demand*. Furthermore, there is given a single *link type* with positive integral *capacity* k .

A *geometric network* is a directed, weighted (finite) *multigraph* embedded in \mathbb{E}^2 . The *cost* of a geometric network G , denoted $c(G)$, is the sum of the lengths of all links used by G , where in the sum we count all copies of parallel edges (multiedges).

A geometric network G is *feasible* with respect to a given set of sources and sinks with demands, and the link type, if the following conditions hold:

- the vertex set of G contains all sources $\{s_1, \dots, s_{n_s}\}$, all sinks $\{t_1, \dots, t_{n_t}\}$, and potentially other vertices (called *Steiner vertices*),
- each copy of a multiedge in G is a link of capacity k ,
- there is a flow in G from the sources to the sinks that is consistent with the directions of used links, saturates the demands of the sources and the sinks, and never exceeds the capacity k in a link.

If G is feasible then one can find such a flow in polynomial time using standard algorithms for the maximum-flow problem with multiple sources and multiple sinks. Furthermore, since the demands and the link capacity are assumed to be integral, we can presume, without loss of generality, that the flow found is integral.

The objective of the *capacitated geometric network design (CGND) problem* is to construct a feasible geometric network that minimizes the total length of all links (i.e., all copies of multiedges) used. If the set of sinks is a singleton then the problem is termed as the *single-sink capacitated geometric network design (SCGND) problem*.

3 Bounding the Number of Steiner Points in Optimal Solution

In this section we consider the CGND problem in \mathbb{E}^2 with sources and sinks having positive integral demands. If multiple sources or sinks are in the same location, we merge them to create a single source or sink. We shall show that the number of Steiner points in an optimal solution can be upper bounded by a polynomial function of D .

For this purpose, we will consider a special class of multigraphs that are feasible solutions to the CGND problem, which we call *minimizers*. We will show that for any $\varepsilon > 0$ there is a minimizer that gives a $(1 + \varepsilon)$ -approximation to the CGND instance. We then analyze geometric properties of the minimizers, and show that *each* minimizer has a special geometric structure, namely it can have only three types of Steiner vertices. We then define an operation of shifting a cycle in a minimizer. We show that this operation transforms a minimizer into another minimizer without increasing the cost, can be performed on a minimizer only a finite number of times, and when the operation cannot be performed any more, the resulting minimizer has a small number of Steiner vertices. We get that for any $\varepsilon > 0$ there *exists* a minimizer that gives a $(1 + \varepsilon)$ -approximation to the CGND instance and has a small number of Steiner vertices. Then we can show that there is an optimal solution which has a small number of Steiner vertices.

We will consider feasible multigraphs together with associate integral flows certifying the feasibility of the multigraphs. For a multigraph $M = (V, E)$ we define the *value* of M as $\text{val}(M) = \sum_{v \in V} (\deg(v) - 2)$, where the degree of a vertex is counted with multiplicities in the case of multiple edges. Among the solutions with the same cost, we will prefer the one with a smaller value.

Without increasing the cost or the value we can modify the multigraph M and an associated integral flow f in M to satisfy the following properties:

- there are no isolated vertices in M ,
- all Steiner vertices in M have degree at least 3 and are contained in the smallest square containing all sources and sinks (to satisfy this property we move each Steiner vertex from outside the square to the closest point on the boundary),
- if two vertices of M are coincident, then there is no edge between them (otherwise we merge them into one vertex),
- each edge of M is used by f ,
- the amount of flow entering and leaving each vertex is at most D .

For technical reasons, and without loss of generality, in this section we consider only pairs (M, f) satisfying the above properties. We get that the degree of a vertex is upper-bounded by $2D$. The Steiner vertices have degree at least 3, and the number of non-Steiner vertices is not greater than $2D$ (i.e. they contribute at least $-2D$ to $\text{val}(M)$), therefore the number of Steiner vertices in M is not greater than $\text{val}(M) + 2D$. Since $2|E| = \sum_{v \in V} \deg(v)$, we get that $|E| = \frac{1}{2} \text{val}(M) + |V| \leq \frac{3}{2} \text{val}(M) + 4D$.

For any problem instance, $v \in \mathbb{N}$ and ε , let $\mathcal{M}_v^\varepsilon$ denote the set of all multigraphs that together with some flow give a $(1 + \varepsilon)$ -approximate solution with a value at most v .

Lemma 1. *For any problem instance, $\varepsilon > 0$ and $v \in \mathbb{N}$ we can equip the set $\mathcal{M}_v^\varepsilon$ with a metric δ such that the metric space $(\mathcal{M}_v^\varepsilon, \delta)$ is compact, and the network cost is a continuous function in $(\mathcal{M}_v^\varepsilon, \delta)$.*

Definition 1. *A multigraph M is a minimizer if it is a feasible solution to the CGND problem and if there does not exist another feasible solution with a smaller cost and not greater value or with the same cost and a smaller value.*

Lemma 2. *For any $\varepsilon > 0$ and any instance of the CGND problem, there is a $(1 + \varepsilon)$ -approximate solution which is a minimizer.*

Proof. Consider any multigraph M which gives a $(1 + \varepsilon)$ -approximate solution for the given instance. It has some finite value v . Let $(\mathcal{M}_v^\varepsilon, \delta)$ be the metric space from Lemma 1. The metric space is compact and the cost is a continuous function in it, therefore there exists a multigraph with a minimum cost in $\mathcal{M}_v^\varepsilon$. The multigraph with a minimum value amongst all multigraphs with the minimum cost in $\mathcal{M}_v^\varepsilon$ is the desired minimizer. \square

In the remaining part of this section we will prove the following theorem.

Theorem 1. *For any $\varepsilon > 0$ and an instance of the CGND problem there is a minimizer M which is a $(1 + \varepsilon)$ -approximate solution and for which $|S| + |Z| \leq 32D^4 + 4D^2$, where S is the set of Steiner vertices of M and Z is the set of points on the plane where the edges of M cross without a Steiner vertex.*

Theorem 1, together with Lemma 1, yields the main result of this section.

Theorem 2. *Any instance of the CGND problem has an optimal solution M_{OPT} which satisfies the following properties: the number of Steiner vertices is at most $32D^4 + 4D^2$, there are no two coincident vertices, the degree of each vertex is at most $2D$ and the edges cross only in vertices.*

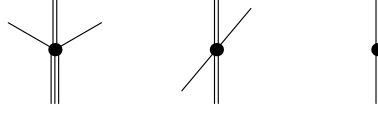


Fig. 2. A real vertex, a crossing and an optional vertex (all for $c = 2$)

3.1 Vertex Types of a Minimizer

We prove here that there can be only three types of Steiner vertices in a minimizer.

Definition 2. Let $M = (V, E)$ be a multigraph with a flow f . A vertex $v \in V$ is called a real vertex if there is an integer $c \geq 0$ and a direction $\alpha \in [0, 2\pi)$ such that:

- the edges incident with v are exactly: c edges with the direction α , $c + 1$ edges with the direction $\alpha + \pi$, and two single edges with the directions $\alpha \pm \pi/3$ (see Fig. 2),
- the group of $c + 1$ edges is directed in one way (out of the vertex or into the vertex), and all the remaining edges are directed in the opposite way,
- the flow in any of the single edges is smaller than the flow in any of the $c + 1$ edges.

Definition 3. Let $M = (V, E)$ be a multigraph with a flow f . A vertex $v \in V$ is called a crossing if there is integer $c \geq 1$ and directions $\alpha, \beta \in [0, 2\pi)$ s.t. $\beta \neq \alpha, \alpha + \pi$ and:

- the edges incident with v are exactly: c edges with the direction α , c edges with the direction $\alpha + \pi$, and single edges with the directions β and $\beta + \pi$ (see Fig. 2),
- all the c edges in one group are directed in one way, the directions of the two groups of edges are opposite, the directions of the two single edges are opposite,
- if $c \geq 2$ then the flow in any single edge is smaller than the flow in any of the non-single edges with the opposite direction.

Definition 4. Let $M = (V, E)$ be a multigraph with a flow f . A vertex $v \in V$ is called an optional vertex if there is an integer $c \geq 2$ and a direction $\alpha \in [0, 2\pi)$ such that:

- the edges incident with v are exactly: c edges with the direction α and c edges with the direction $\alpha + \pi$ (see Fig. 2),
- all the c edges in one group are directed in one way, the directions of the two groups of edges are opposite.

We specify some properties which the Steiner vertices of a minimizer must fulfill.

Property 1. Let $M = (V, E)$ be a minimizer, $v \in V$, and $e_1, e_2 \in E$ be edges incident with v and directed in opposite ways. Then the angle between e_1 and e_2 is at least $2\pi/3$.

Property 2. Let $M = (V, E)$ be a minimizer with a flow f . Let $v \in V$ and let $e_1, e_2 \in E$ be edges incident with v and directed in the same way. If the angle between e_1 and e_2 is smaller than $2\pi/3$, then $f(e_1) + f(e_2) > k$.

Property 3. Let $M = (V, E)$ be a minimizer with a flow f . Let $v \in V$ and let $e_1, e_2, e_3 \in E$ be edges incident with v contained in an open halfplane with the borderline passing through v . If e_1 and e_2 are directed in the same way and opposite to the direction of e_3 , then $f(e_1) + f(e_2) - f(e_3) > k$ and, in particular, $f(e_3) < f(e_1), f(e_2)$.

For a vertex v we call an edge e incident with v *incoming* (*outgoing*) if it is directed towards v (out of v). Using Properties 1–3 we can show the following lemma:

Lemma 3. *Let $M = (V, E)$ be a minimizer. If $v \in V$ is a Steiner vertex such that both incoming and outgoing edges have multiple directions, then v is a crossing. On the other hand, if $v \in V$ is a Steiner vertex such that all outgoing (or incoming) edges have the same direction $\alpha \in [0, 2\pi)$, then v is a real vertex or an optional vertex.*

This immediately implies the following theorem.

Theorem 3. *Each Steiner vertex in a minimizer is a real vertex, a crossing or an optional vertex.*

3.2 Graph Analysis and Cycle Argument

From Theorem 3 we know that a Steiner vertex in a minimizer is a real vertex, a crossing or an optional vertex. We can easily remove optional vertices from a minimizer.

Lemma 4. *For any minimizer M there is a corresponding minimizer M' with the same cost, value, number of real vertices and crossings and with no optional vertices.*

In this section we will prove Theorem 1. We introduce a procedure modifying minimizers to decrease the number of real vertices. We also show, that if we cannot decrease the number of real vertices any further, the total number Steiner vertices must be small.

Definition 5. *Let M be a minimizer without optional vertices. A cycle \mathcal{C} in M is called a Steiner cycle if:*

- \mathcal{C} passes only through Steiner vertices,
- if \mathcal{C} passes through a crossing vertex v , then \mathcal{C} does not change direction in v ,
- if \mathcal{C} passes through a real vertex v , then \mathcal{C} either does not change the direction, or changes it by $\pi/3$,
- \mathcal{C} does not pass through an edge more than once.

Lemma 5. *Let M be a minimizer without optional vertices, S the set of Steiner vertices of M and \mathcal{Z} the set of points on the plane where the edges of M cross without a Steiner vertex. If M has no Steiner cycle, then $|S| + |\mathcal{Z}| \leq 32D^4 + 4D^2$.*

We introduce an operation of *shifting* a Steiner cycle \mathcal{C} by a distance $\Delta > 0$ in a minimizer M with no optional vertices. We orient \mathcal{C} in one of the two possible directions and shift each edge of \mathcal{C} by a distance Δ to the left according to the direction chosen. The real vertices where \mathcal{C} changes direction are moved to the intersection points of the shifted edges (see Fig. 3). A Steiner cycle has no repeating edges, so we never have to shift an edge in two directions at once.

Moving the real vertices as described above requires us to make some other modifications to the graph. We do not want to shift or change directions of edges not belonging to \mathcal{C} — they can only become longer or shorter. To achieve this, in some cases instead of moving a real vertex we split the vertex into a real vertex of degree 3 and a crossing, and then only move the vertex of degree 3 to the required position. We also move the

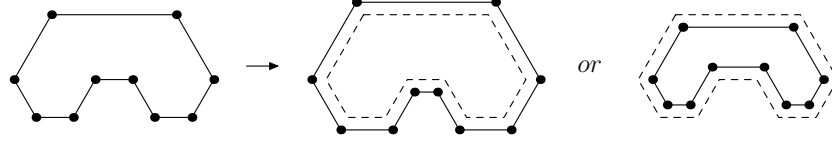


Fig. 3. Shifting a Steiner cycle. Only the real vertices on which the cycle changes direction have been pictured. The obtained cycle depends on the orientation of the original cycle.

real vertices and crossings of \mathcal{C} , on which \mathcal{C} does not change the direction, possibly after splitting the vertices in two. The case analysis shows that shifting a Steiner cycle neither changes the cost and the value of the minimizer nor the number of the real vertices. The resulting graph has a feasible flow and is a minimizer.

When a vertex from \mathcal{C} hits a vertex it is incident with, we merge the two vertices. One can show that the shifting operation can be performed until one of the following happens: a vertex from \mathcal{C} gets merged with a non-Steiner vertex or two real vertices get merged (e.g. when a cycle edge gets contracted to a single point). The resulting multigraph has either a smaller number of real vertices, or the same number of real vertices and a larger sum of degrees of non-Steiner vertices.

Proof (of Theorem 1). Fix an $\varepsilon > 0$. By Lemma 2, there is a $(1 + \varepsilon)$ -approximate solution which is a minimizer. Let M_0 be such a minimizer that also minimizes the number of real vertices — we cannot decrease the number of real vertices of M_0 without increasing the cost or the value of M_0 . Let M_1 be the minimizer obtained from M_0 by removing the optional vertices, as in Lemma 4. The number of real vertices, the cost and the value of M_1 is the same as in M_0 .

Suppose that M_1 has a Steiner cycle. We shift the cycle as far as possible. As the result, we obtain a minimizer with the same cost and value, where the shifted cycle is not a Steiner cycle any more, and the resulting multigraph has the same number of real vertices and a larger sum of degrees of non-Steiner vertices (any other result of shifting a cycle would decrease the number of real vertices, and that cannot happen for M_1).

As long as M_1 has some Steiner cycle left, we shift it as far as possible. We can perform this operation only a finite number of times, as each time the sum of the degrees of the non-Steiner vertices increases, and in a minimizer it is upper bounded by $4D^2$. At some point there will be no Steiner cycle left. The graph obtained from M_1 will be a minimizer with no Steiner cycles, and from Lemma 5 we know that in such a minimizer $|S| + |\mathcal{Z}| \leq 32D^4 + D^2$. \square

4 Quasi-Polynomial Time Approximation Scheme

In this section we consider the CGND problem with the total demand $D \leq n^C$ for some constant C . We will design a quasi-polynomial time approximation scheme for the above problem that works for any capacity k . We can assume that $k \leq D$.

From Theorem 2, we know that there is an optimal network $M_{OPT} = (V, E)$ that satisfies the following properties: the number of Steiner vertices is at most $32D^4 + 4D^2$, there are no two coincident vertices, the degree of each vertex is at most $2D$ and the

edges cross only in vertices. We have $|V| \leq 38n^{4C}$, $|E| \leq 38n^{5C}$. These bounds make it possible to use the Arora's framework [1] for the TSP problem. Due to lack of space, we give only an overview of the algorithm.

Dividing into subproblems. We can prove the following lemma.

Lemma 6. *In polynomial time, we can partition the input points such that every optimal solution consists of the disjoint solutions for the points from each partition set, and the cost of the optimal solution for the points from any partition set P_i is at least $\frac{\ell_i}{nk}$, where ℓ_i is the side length of the smallest square containing all points from P_i .*

We divide the original problem instance into a collection of at most n independent instances, as given by Lemma 6. In the rest of this section we can therefore assume that all the input points are inside some square of size $\ell \times \ell$, and $c(M_{OPT}) \geq \frac{\ell}{nk}$.

Perturbation. We divide the square $\ell \times \ell$ into *unit squares* using a uniform grid of size polynomial in n and D . We modify the problem by shifting all input points to the nearest gridpoint (intersection of grid lines), and merging coincident sinks and sources.

Let M'_{OPT} be a multigraph obtained from M_{OPT} by this operation. Notice that the edges in M'_{OPT} cross only in Steiner vertices, similarly as the edges of M_{OPT} . Using the upper bounds on the number of vertices and edges in M_{OPT} we can show:

Lemma 7. *If the size of the grid is at least $\frac{80}{\varepsilon} \cdot n^{6C+1} \times \frac{80}{\varepsilon} \cdot n^{6C+1}$, we have $c(M'_{OPT}) \leq (1 + \varepsilon) \cdot c(M_{OPT})$.*

Randomized dissection and portals. We make a randomized dissection of the square — a recursive partitioning of the square into four equal squares, performed until we obtain the unit squares. The size of the grid is the smallest power of 2 which is at least $2 \cdot \frac{80}{\varepsilon} \cdot n^{6C+1}$. Also, instead of starting with the original $\ell \times \ell$ square, we start with a $2\ell \times 2\ell$ square with the original square shifted randomly inside. A more detailed description of the randomized dissection can be found in [1,2]. The number of levels of the dissection is $\mathcal{L} = O(\log(2 \cdot \frac{80}{\varepsilon} \cdot n^{6C+1})) = O(C \cdot \log \frac{n}{\varepsilon})$.

Similarly as in [1,2], we create $m = \frac{2\mathcal{L}}{\varepsilon} = O(\frac{C}{\varepsilon} \log \frac{n}{\varepsilon})$ equidistant portals on the boundary of each square in such a way that the portals are in all corners of the square, where m is a power of 2. We modify the solutions to the CGND problem in such a way, that the edges can cross the boundaries of the squares only in portals (i.e. we create additional Steiner points in portals). Notice that all the input points are in portals. We do not allow any Steiner points outside of portals, and we do not allow the edges to cross outside of the portals. In the remaining part of this section we shall present a quasi-polynomial time algorithm that finds an optimal solution to the modified problem. The QPTAS follows, as the following hold:

Lemma 8. *The optimal solution to the modified problem has expected cost at most $(1 + \varepsilon)^2 \cdot c(M_{OPT})$.*

Lemma 9. *From the optimal solution to the modified problem we can get a solution for the original problem with an expected cost at most $(1 + \varepsilon)^3 \cdot c(M_{OPT})$.*

Dynamic programming. We want to find an optimal solution to the modified problem, i.e., where all sinks and sources are in portals, the Steiner points can be created only in portals and the edges do not cross outside of the portals. We can consider only solutions (multigraphs with a specified flow) where: the flow through each portal goes in one direction only (into or out of the square) and the amount of it is at most D .

We find the optimal solution using dynamic programming. The interface of a square specifies the amount of flow and the direction of it (into or out of the square) for each portal. The number of interfaces of a square is $(1 + 2D)^m = n^{O(C^2 \varepsilon^{-1} \cdot \log(n/\varepsilon))}$, which is quasi-polynomial in n . For each square and for each interface we want to find the solution with the minimum cost satisfying the interface (if there is a feasible solution for a given interface). Since we have doubled the size of the original square enclosing the input points, we may assume that no input points are on the perimeter of this square. Hence, the solution to the problem is the minimum cost solution for the doubled original square with an empty interface. We start with the unit squares:

Lemma 10. *The minimum cost solutions to the leaf-subproblems, i.e., to the interfaces of the unit squares can be found in time $n^{O(C^2 \varepsilon^{-1} \log(n/\varepsilon))}$.*

The idea of the proof is as follows. We consider only solutions where edges do not cross inside the squares, i.e. they go along some triangulation of the square, where the vertices of the triangulated polygon are the portals. For each triangulation we consider multigraphs consistent with the triangulation, compute their cost and check whether the flow required by the interface can be transferred through the multigraph.

Lemma 11. *The minimum cost solutions to the remaining subproblems, i.e., to the interfaces of the non-unit squares can be found in time $n^{O(C^2 \varepsilon^{-1} \log(n/\varepsilon))}$.*

By applying Lemmas 9–11, we obtain our main theorem in this section.

Theorem 4. *For any $\varepsilon > 0$, there is a randomized $n^{O(C^2 \varepsilon^{-1} \log(n/\varepsilon))}$ -time algorithm for capacitated geometric network design with a total of n sources and sinks and total demand upper bounded by n^C , which yields a solution whose expected cost is within $(1 + \varepsilon)$ of the optimum. The algorithm can be derandomized similarly as those in [1,3].*

Remark 1. An analogous theorem holds when instead of the requirement on the total demand of sinks balancing the demand of sources, the sinks have unlimited demand.

5 Polynomial-Time Approximation Scheme for Single Sink

In this section we consider the SCGND problem, with the input consisting of a single sink s and a set P of n sources with a total demand $D \leq n^C$. The capacity of the edges is k , and throughout this section we will assume that $k \leq 2^{O(\sqrt{\log n})}$. We will construct a polynomial-time approximation scheme for this problem.

The idea of the algorithm is as follows. We partition the set of sources into two subsets, depending on their distance from the sink. The "outer" set has a total demand upper bounded by a polynomial function of k . Using the algorithm from Section 4 we construct a near-optimal network which allows to transfer the flow from the outer

sources to some Steiner vertices closer to the sink. The Steiner vertices now become new sources. As the number of points and the total demand is much smaller than n , the running time is polynomial in n . We then solve the problem for the "inner" sources together with the new sources. We show that for such a set of points the cost of the minimum TSP tour is small compared to the cost of the optimal solution. That allows us to use a simple algorithm to find a near-optimal solution. Combining the two networks together gives a near-optimal solution to the SCGND problem.

Let L be maximum distance between a source and the sink, i.e. $L = \max_{p \in P} d(p, s)$. Let OPT be an optimal solution (network) for the SCGND problem, and TSP be the shortest traveling salesman tour for $P \cup \{s\}$. The following lemma can be proved similarly as Theorem 1 in [5] for the capacitated vehicle routing problem.

Lemma 12. *There is a polynomial time algorithm that outputs a solution with cost at most $(1 + \varepsilon) \cdot c(\text{TSP}) + \frac{1}{k} \sum_{p \in P} d(p, s) \cdot \mathfrak{d}(p)$. Furthermore, the following hold:*

$$\frac{1}{k} \sum_{p \in P} d(p, s) \cdot \mathfrak{d}(p) \leq c(\text{OPT}) \leq c(\text{TSP}) + \frac{1}{k} \sum_{p \in P} d(p, s) \cdot \mathfrak{d}(p) .$$

If $c(\text{TSP}) < \frac{\varepsilon}{k} \sum_{p \in P} d(p, s) \mathfrak{d}(p)$, then Lemma 12 gives a polynomial-time $(1 + \varepsilon)$ -approximation. Therefore from now on we assume that $c(\text{TSP}) \geq \frac{\varepsilon}{k} \sum_{p \in P} d(p, s) \mathfrak{d}(p)$.

Working with this assumption, we use the following lemma to upper bound the sum of the distances from the sources to the sink:

Lemma 13. *If $c(\text{TSP}) \geq \frac{\varepsilon}{k} \sum_{p \in P} d(p, s) \cdot \mathfrak{d}(p)$ then there is a constant c such that*

$$\sum_{p \in P} d(p, s) \cdot \mathfrak{d}(p) \leq \left(\frac{c \cdot k}{\varepsilon} \right)^2 \cdot L .$$

Consider a closed disk \mathcal{Q} of radius αL with the center at the sink s , where we set $\alpha = \frac{\varepsilon^4}{c^4 \cdot k^2}$. Let P_{out} be the subset of points from P that are outside the disk \mathcal{Q} (i.e., for which $d(p, s) > \alpha L$), and let $P_{in} = P \setminus P_{out}$. By Lemma 13, we have $\sum_{p \in P_{out}} d(p, s) \cdot \mathfrak{d}(p) \leq \left(\frac{c \cdot k}{\varepsilon} \right)^2 \cdot L$, and hence $\sum_{p \in P_{out}} \mathfrak{d}(p) \leq \left(\frac{c \cdot k}{\varepsilon} \right)^2 \cdot \frac{1}{\alpha}$.

We partition the optimal network into two parts: OPT_{in} and OPT_{out} , which consist of the edges lying respectively inside and outside \mathcal{Q} (we add artificial Steiner points on the boundary of \mathcal{Q} to divide the edges). Then $c(\text{OPT}) = c(\text{OPT}_{in}) + c(\text{OPT}_{out})$.

Lemma 14. *In time $k^{O(\varepsilon^{-1} \log k)}$ we can find a set P'_{out} of $\sum_{p \in P_{out}} \mathfrak{d}(p)$ points on the boundary of the disk \mathcal{Q} and a network of cost at most $c(\text{OPT}_{out}) + 3 \cdot \varepsilon \cdot c(\text{OPT})$ that allows sending all the flow from P_{out} to the unit sinks P'_{out} .*

The idea of the proof is as follows. We create equidistant points (portals) on the boundary of the disk \mathcal{Q} and treat them as sinks with infinite capacity. If the number of the portals is large enough (but still smaller than the total demand of the points from P_{out}), the cost of an optimal network which allows to send the flow from the sources P_{out} to the created sinks is not greater than $c(\text{OPT}_{out}) + \varepsilon \cdot c(\text{OPT})$. A good approximation of such a network can be found by the algorithm from Theorem 4 adapted to Remark 1.

Lemma 15. *In polynomial time we can find a network which sends the flow from the points $P_{in} \cup P'_{out}$ to the sink s with cost at most $c(\text{OPT}_{in}) + 2 \cdot \varepsilon \cdot c(\text{OPT})$.*

One can prove it by showing that the algorithm from the Lemma 12 for the points $P_{in} \cup P'_{out}$ gives the desired approximation ratio. Combining the algorithms from Lemmas 12, 14 and 15 gives us a solution to the original SCGND problem with cost at most $(1 + 5\varepsilon) \cdot c(\text{OPT})$ and running in time $k^{O(\varepsilon^{-1} \log k)} + n^{O(1)}$. This gives our main result.

Theorem 5. *For any $\varepsilon > 0$, there is a deterministic algorithm for the SCGND problem with total demand polynomial in n and edge capacity k , which runs in time $k^{O(\varepsilon^{-1} \log k)} + n^{O(1)}$ and yields a solution whose cost is within $(1 + \varepsilon)$ of the optimum. For $k = 2^{O(\sqrt{\log n})}$, it runs in polynomial time.*

6 Final Remarks

It is an intriguing question of whether or not our upper bound on the number of Steiner points in an optimal solution to CGND could be extended to include several types of links. Such an extension would lead to corresponding extensions of our QPTAS for CGND and our PTAS for SCGND. Also, the questions if one can get a PTAS for SCGND for larger k and a PTAS for CGND for some values of k are interesting.

References

1. Arora, S.: Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM* 45(5), 753–782 (1998)
2. Czumaj, A., Czyzowicz, J., Gąsieniec, L., Jansson, J., Lingas, A., Zylinski, P.: Approximation algorithms for buy-at-bulk geometric network design. In: Dehne, F., Gavrilova, M., Sack, J.-R., Tóth, C.D. (eds.) *WADS 2009*. LNCS, vol. 5664, pp. 168–180. Springer, Heidelberg (2009)
3. Czumaj, A., Lingas, A.: On approximability of the minimum-cost k -connected spanning subgraph problem. In: *Proc. 10th SODA*, pp. 281–290 (1999)
4. Garey, M.R., Johnson, D.S.: *Computers and Intractability. A Guide to the Theory of NP-completeness*. W.H. Freeman and Company, New York (1979)
5. Haimovich, M., Rinnooy Kan, A.H.G.: Bounds and heuristics for capacitated routing problems. *Mathematics of Operation Research* 10(4), 527–542 (1985)
6. Hassin, R., Ravi, R., Salman, F.S.: Approximation algorithms for a capacitated network design problem. *Algorithmica* 38, 417–431 (2004)
7. Hwang, F.K., Richards, D.S., Winter, P.: *The Steiner Tree Problem*. *Annals of Discrete Mathematics*, vol. 53. North-Holland, Amsterdam (1992)
8. Mitchell, J.S.B.: Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k -MST, and related problems. *SIAM Journal on Computing* 28(4), 1298–1309 (1999)
9. Morsy, E., Nagamochi, H.: Approximation to the minimum cost edge installation problem. In: Tokuyama, T. (ed.) *ISAAC 2007*. LNCS, vol. 4835, pp. 292–303. Springer, Heidelberg (2007)
10. Salman, F.S., Cheriyan, J., Ravi, R., Subramanian, S.: Approximating the single-sink link-installation problem in network design. *SIAM J. Optimization* 11(3), 595–610 (2000)
11. Zachariasen, M.: A catalog of Hanan grid problems. *Networks* 38(2), 76–83 (2001)